APPENDIX B

EXPRESS MAIL LABEL
NO.: EL386270487US

Appendix B will be understood and appreciated from the following detailed description, taken in conjunction with the drawings in which:

Fig. 54 is a simplified illustration of a single snapshot image (referred to herein is a "Snap") covering the area around a single Color_defect report, constructed and operative in accordance with a preferred embodiment of the present invention;

Fig. 55 is a simplified illustration of snaps covering the area around multiple Color_defect reports, constructed and operative in accordance with a preferred embodiment of the present invention; and

Fig. 56 is a simplified schematic illustration of the structure of a line of 48 snap reports, constructed and operative in accordance with a preferred embodiment of the present invention.

A Snap is a rectangular or modular-rectangular region that contains RGB (red, green, blue)-information of the pixels of that region. The RGB-information is collected from a certain area that surrounds a report of the type Color_defect. The minimal region in which the RGB data is collected around a single report is of size n x n (pixels). In the case of the present embodiment n = 48.

Reference is now made to Figure 54 which is an illustration of a single snap covering the area around a single Color_defect report, which is useful in the understanding of a preferred embodiment of the present invention.

Single snap report

The single snap report is a 32-bit word. Its internal structure is the following:

8 bit : x_s - x coordinate in 8-bit representation.

8 bit : r - value of red.

8 bit : g - value of green.

8 bit : b - value of blue.

The 8-bit representation of the x-coordinate is faithful provided that the snap reports in a given line are ordered by their 12-bit value of the x-coordinate. The conversion from the 8-bit representation of the x-coordinate to a 12-bit representation is explained below.

The origin and structure of snap data

Snaps are typically created “on-line”, during a scan, for example during the inspection scenario. Each slice is treated separately and is divided into basic block units. The basic unit is a square of size 16 x16. Each Color_defect report leads to the recording of 9 blocks of RGB data: the block which contains the Color_defect report itself and its 8 neighboring blocks, as illustrated in Fig. 54.

A snap area that is recorded due to several Color_defect reports, is determined according to the following rules:

1. The minimal snap is typically of size 48 x 48 $(\text{pixels})^2$, and it consists of nine blocks of size 16 x 16 $(\text{pixels})^2$ each.

2. Each snap report preferably appears only once, even if it was originated due to several Color_defect reports (see Figure 55).

Snap reports are typically ordered in lines, according to their y and x coordinates. Each line contains a sequence of 16*N single snap reports, accompanied by information about the slice in which the reports were originated, and the number of the reports in the line. The reports in the are divided into N groups of 16 reports all of which have the same value of x in the 8-bit representation. Fig. 56 shows schematically the structure of the line of snap reports, with the minimal value of N (N=3).

The 8-bit to 12-bit conversion of the x coordinate

The conversion from the 8-bit representation to the full 12-bit representation of the x coordinate is done in the following manner:

• Let x_s be the x-coordinate in the 8-bit representation of a given snap report.

• Let n_b be the number of the snap report within the relevant group of 16 reports, all carrying the same 8- bit value of x_s. The numbering is done in the following way: the first report in the group has $n_b = 0$, the second $n_b=1$, etc.

Then the 12-bit representation of the snap report’s x-coordinate is given by:

$$x = x_s * 16 + n_b$$

Ds_array<Snap>

Ds_array<Snap> contains Snap reports ordered by their y and x coordinates. The structure of the single Snap report is as mentioned earlier. The x coordinate is given in

8-bit representation, and is preferably converted to a 12- bit representation. As in the case of all data sources, a pointer to the beginning of each ds_array_line is available, as well as the information about the slice in which the data source was originated .